Numerical control system based on a programmable logic device

Grzegorz Śmigielski[0000-0003-3781-0894]
Kazimierz Wielki University in Bydgoszcz, Poland

Abstract. The article presents a numerical control system of a multi-axis machine built using a programmable logic device. The system consists of PC, motor controller (based on FPGA), power stages and motors. The complete code of controller was written in the VHDL language and implementing in Xilinx Spartan 3 board. The PC program was created using LabVIEW. The logical and functional tests of the system have been carried out. The application of a programmable device enables its quick configuration according to the requirements set by the user. The main advantage of FPGA is the option to expand to following modules, such as the incremental encoder support module, PWM, FOC etc. In these applications, the advantage of the programmable system becomes visible, which due to its specifics, is built for parallel processing or generating signals.

1 Introduction

The necessity of precise control of movement and position parameters of a specified object is applicable to all modern process machines, such as numerical machines (e.g. 3D printers) or robots [1, 2]. The possibility of completing remarkably precise movement enables building fully automated manufacturing lines. Some benefits of automation are reduced internal costs of manufacturing, increased operation efficiency, improved product quality, decreased labour intensity and nuisance, and higher safety. In addition to the ever increasing requirements regarding the precision and control range of both position and speed, as well as movement dynamics, expectations also increase with regards to the flexibility of the means of productions, understood as the possibility of easy and relatively quick adjustment to changing manufacturing tasks[3, 4, 5, 6]. Combining all these aspects is very difficult and it would even be impossible without the achievements of the microelectronic industry.

There are many different architectures of numerical control systems that depend on the components in use [1]. Systems based on PCs are very popular due to their availability and low price. They consist of one computer, a controller and motors. The controller can be built based on a microprocessor system, programmable system etc. Most controllers of drive systems use microprocessors or DSP processors[7, 8]. This article shows the example of a controller built using a programmable FPGA system of the Spartan series by Xilinx (XC3S500E).
2 CNC system design

The layout of a CNC system built using stepping motors is shown in figure 1.

![Fig. 1. CNC system built using stepping motors.](image)

The PC with proper software combines an information input (and output) unit and a computing unit. Information on the movement of the object is input to the PC (e.g. file with data in language ‘G’), converted to numerical data on the number of unit steps in the given time interval $\Delta t$ to be completed by the given actuator. The controller supervised by the PC is designed to generate impulses to control the motors on specific axes. It is important that they are evenly distributed along a time section $\Delta t$. Currents with a suitable value are generated by power stages. The controller enables operating motors in the microstep mode. The controller has:

- outputs for the direct control of 4 power stages, with a simple extension option;
- 16 discrete inputs;
- 8 discrete outputs;
- analogue outputs with adjustable voltage;
- RS-232C serial transmission system, which provides communication with the master PC.

The control PC sends 9-byte data packages to the controller. They contain the number of broadcast byte package, command for the controller (interpolation, data zeroing, turn off), position increments for 4 axes (motors), the condition of discrete outputs and the analogue output, the checksum. The controller, after checking the correctness of the data received from the PC, returns 8-byte packages with information on transmission correctness, the number of the received byte package, the number of the read byte package, the state of discrete inputs(2 bytes), information on the free space in the reception buffer, controller status and checksum.

To complete the controller design, Xilinx ISEWebpack software was used. The complete code was written in the VHDL language[9, 10, 11]. The most important blocks of the controller are:

- The serial transmission transmitter/receiver – the serial transmission system was implemented in the form of two state systems.
- Data write/read block.
The data received (7 of 9 bytes) is written to internal memory. It was implemented using RAM blocks, configured as two-port memory units with a capacity of 256 16-bit words.

- Motor control block.
  The step motor microstep control consists in changing the amplitude of currents in phases, according to the following dependencies:

\[
\begin{align*}
I_A &= I_{\text{max}} \times \sin \theta_S \\
I_B &= I_{\text{max}} \times \cos \theta_S
\end{align*}
\]  

In the digital system, the sinus (and cosinus) functions can be tabled in the form of discrete values. If the number of points, at which calculations are made, is sufficient, a quasi-continuous current value variation is obtained. The required values are as follows:

\[
\theta_S = \frac{2\pi \cdot n}{m}
\]  

where:
- \( n \) – microstep number (1 ... \( m \)), \( m = 4 \times \text{number of microsteps per full step} \);
- \( I_A \) – A-phase current, \( I_B \) – B-phase current, \( I_{\text{max}} \) – maximum current of the given phase.

To the prototype board of the controller, through expansion connectors, a board with LMD18245 power systems by National Semiconductor are connected to control the operation of motors. To supply one motor, two stages like that are needed (one per phase). Microstep control with step division into 16 parts and fixed torque requires generating 16 non-zero current values. In the controller built, current values in motor phases are selected so that in microstep operation with step division into 16 parts, the least possible torque fluctuations (\( \leq 6\% \)) and the least possible rotor setting error, in relation to motor supply with a perfect course current (\( \leq 27\% \)), are obtained. The maximum current value in A (the nominal current of the motor) is obtained through formula:

\[
I = (V_{\text{ref}} \times \frac{D}{16}) \times (250 \times 10^{-6} \times R_1)
\]  

where: \( V_{\text{ref}} \) – reference voltage [V], \( D \) – maximum value fed by the control system to binary inputs M1–M4 (0 to 15), \( R_1 \) – resistance of the setting resistor.

Based on the above dependency, a change in D causes a change in the current flowing through the motor windings. The control of the applied power stage consists in feeding the right current and direction values to the binary inputs. It occurs as follows: the controller downloads information on the number of steps to complete in time \( \Delta t \) from the data memory and calculates time segments \( \tau_{\text{int}} \) between the subsequent movements (steps). The current value is changed according to the values written in the ROM of the programmable system. For each motor phase, 2 ROM tables are reserved with a capacity of 64 5-bit words. The four younger bits (M1 to M4) define the current value, while the oldest bit defines the direction. Every subsequent step increases (rotor rotates to the right) or decreases (rotates to the left) the indicator value for the table – thus the current value read (and fed to power system inputs) is different (fig.2).
Motor control is achieved in four separate process instructions of the VHDL language, executed every 1 μs.

Table 1 shows the summary of use of the logical resources of the system. The total number of equivalent gates is 178990.

<table>
<thead>
<tr>
<th>Item name</th>
<th>Number of items used</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLICE blocks</td>
<td>885</td>
</tr>
<tr>
<td>LUT logical tables in SLICE blocks</td>
<td>1323</td>
</tr>
<tr>
<td>Triggers in SLICE blocks</td>
<td>584</td>
</tr>
<tr>
<td>Input/output cells</td>
<td>78</td>
</tr>
<tr>
<td>Triggers in input/output cells</td>
<td>16</td>
</tr>
<tr>
<td>Global clock inputs</td>
<td>1</td>
</tr>
<tr>
<td>Global inputs</td>
<td>1</td>
</tr>
</tbody>
</table>

To check the operation of the controller, a master program was developed in the LabView environment (fig.3).
Motor control is achieved in four separate process instructions of the VHDL language, executed every 1 μs. Table 1 shows the summary of use of the logical resources of the system. The total number of equivalent gates is $17,8990$.

Table 1. Use of the logical resources of the system.

<table>
<thead>
<tr>
<th>Item name</th>
<th>Number of items used</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLICE blocks</td>
<td>885</td>
</tr>
<tr>
<td>LUT logical tables in SLICE blocks</td>
<td>1323</td>
</tr>
<tr>
<td>Triggers in SLICE blocks</td>
<td>584</td>
</tr>
<tr>
<td>Input/output cells</td>
<td>78</td>
</tr>
<tr>
<td>Triggers in input/output cells</td>
<td>16</td>
</tr>
<tr>
<td>Global clock inputs</td>
<td>1</td>
</tr>
<tr>
<td>Global inputs</td>
<td>1</td>
</tr>
</tbody>
</table>

To check the operation of the controller, a master program was developed in the LabView environment (fig. 3).

The program enables access to all essential functions of the controller. The user can define the operation mode – write to memory, interpolation, zeroing or turn off (reset); rotation direction and number of steps to complete by motors per 2 ms (speed) and the condition of discrete outputs and the analogue output. The application also enables reading discrete input states and writing the packages received in response to file.

The measurements were conducted using a digital oscilloscope by Lecroy, model LT374.

![Fig. 3. Master program.](image)

The measurements were conducted using a digital oscilloscope by Lecroy, model LT374.

![Fig. 4. Plot of voltage changes at power stage outputs – illustration of the PWM generator operation.](image)
The first plot (fig. 4) shows the difference in voltage at outputs OUT1 and OUT2 of the power stage at the set rotation speed of 1 step per 2 ms. The course time is 128 ms and in this time the motor completes 64 steps.

The second plot (fig. 5) made at the same set rotation speed was obtained using the mathematical functions of the oscilloscope. The plot shows an averaged voltage difference at outputs OUT1 and OUT2, which can be identified with the current in motor phase.

3 Conclusion

The solution presented does not use the full capacity of the programmable systems, i.e. the implementation of a microprocessor in its resources, which would enable a simpler execution of the tasks related to calculations, order processing and communication. Other logical resources can be used for simultaneous parallel processing of signals, e.g. generating PWM. Unfortunately, the implementation of the Microblaze processor in Xilinx systems requires a paid version of the software for each system of this manufacturer. As an alternative, the solutions offered by Intel – Altera can be used. MAX10 systems are supported in the free version. The implementation of the same processor in the 10M08DAF256C8GES system included in the prototype board MAXimator uses 2100 equivalent gates.

The controller with the PCs and motors constitutes a full numerical control system for a machine. The application of a programmable system enables quick changes to the hardware and software of the device and its quick configuration according to the requirements set by the user. The power of the controlled motors is limited only by the end stages available on the market and compatible with the controller. The above solution features high simplicity on one hand (small number of items used) and considerable flexibility on the other hand. In the solution presented, the FPGA system takes over the tasks usually fulfilled by a dedicated microcontroller and its main advantage is the option to expand to following modules, such as the incremental encoder support module, PWM signal generation module, FOC module (for PMSM motors) or measurement modules based on delay line systems [8,
In these applications, the advantage of the programmable system becomes visible, which due to its specifics, is built for parallel processing or generating signals.

References